



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/663,455

09/15/2003

David Abu Ghazaleh

RSW920030055US1

2196

46270

7590

06/06/2008

(SAUL-RSW) PATENT DOCKETING CLERK
IBM Corporation (SAUL-RSW) C/O Saul Ewing LLP
Penn National Insurance Tower
2 North Second Street, 7th Floor
Harrisburg, PA 17101

EXAMINER

CHAUHAN, LOREN B

ART UNIT

PAPER NUMBER

2193

MAIL DATE

DELIVERY MODE

06/06/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/663,455	Applicant(s) GHAZALEH ET AL.	
	Examiner Loren Chauhan	Art Unit 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 29 January 2008.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-35 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-35 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to remarks filled on 1/29/2008. Claim 22 have been amended. Claims 1-35 are pending for examination.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1-2, 5-22, 24 and 26-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bailey (US Pat. No. 6,701,513) in view of Kodosky (US PG-Pub. No. 2003/0184580).

4. As per claim 1, Bailey teaches the invention substantially as claimed including a method for graphically representing object oriented programming logic (Abstract lines 10-13), the method comprising the steps of:

(1) providing a plurality of different symbols for use in a diagram of object oriented programming logic, each different symbol representing a different type of object in object oriented programming (402, 402a fig. 4A; col. 7, lines 57-61; Col. 8, lines 24-26);

(3) labeling the symbol with a label descriptive of the object's features so that it is distinguishable from other symbols of the same object type (col. 10, lines 53-55, 58-59, 64-65; col. 11, lines 35-38);

(4) for each object assigned to or defined within the main object, drawing a symbol corresponding to that object and labeling the symbol with a label descriptive of the object's features (col. 10, lines 53-55, 58-59, 64-65; col. 11, lines 35-38); and

(5) drawing a line between each object drawn in step (4) and another object in the graphical representation to which it is assigned or within which it is defined (col. 4, lines 23-26).

5. However, Bailey does not explicitly teach selecting an object as a main object of the logic to be represented in the diagram and drawing a symbol corresponding to the main object.

6. Kodosky in analogous art teaches creating an icon (paragraph [0011] lines 10-12) representing a main program in a hierarchical view of the system (paragraph [0012] lines 9-11, 24-25).

7. It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Bailey and Kodosky because

Kodosky's teaching of main object with symbol would help user to easily understand logic that was described in detail according to specification of the system without creating any confusion.

8. As per claim 2, Bailey teaches the method, further comprising the step of: (6) providing a plurality of additional different symbols for use in the diagram, each of the additional different symbols representing a different an object oriented programming element other than an object (col. 4, lines 23-26).

9. As per claim 5, Bailey teaches the method, wherein the labels comprise text (col. 10, lines 53-55).

10. As per claim 6, Bailey teaches the method, wherein step (5) comprises drawing the line between the object defined in step (4) and another object it is most directly assigned to or is most directly defined within (col. 4, lines 36-40).

11. As per claims 7 and 8, Kodosky teaches a method of visually creating a distributed system design and software programming (paragraph [0001] which can be used in documenting software and to prepare a program specification.

12. As per claim 9, Bailey teaches the method, further comprising the step of: (8) repeating steps (1) - (5) to prepare a plurality of separate diagrams corresponding to separate parts of an overall application (col. 8, lines 14-18).

13. However, does not explicitly teach a first object is the main object appearing in at least a first one of the diagrams and is not a main object appearing in at least a second one of the diagrams.

14. Kodosky teaches a first object is the main object appearing in at least a first one of the diagrams and is not a main object appearing in at least a second one of the diagrams (paragraph [0015]).

15. As per claim 10, Kodosky teaches the method wherein a diagram does not disclose objects assigned to and defined within an object and the diagram does disclose objects assigned to and defined within the object (paragraph [0015]).

16. As per claim 11, Bailey teaches the method, an application-level representation disclosing an overall software system (col. 8, lines 14-18).

17. As per claim 12, Bailey teaches the method, wherein the label identifies as disclosing further details of the object (col. 9, lines 50-53).

18. As per claim 13, Bailey does not explicitly teach wherein the symbols representing different object types include: a symbol for representing objects that are application type objects; a symbol for representing objects that are window type objects; a symbol for representing objects that are class type objects; a symbol for representing objects that are event script type objects; and a symbol for representing objects that are method type objects.

19. Bailey does teach plurality of objects or icons representing text boxes, radio buttons, scroll bars, menu bars and so on (col. 2, lines 3-9; col. 7, lines 57-61; col. 8, lines 8-11, 15-18, 24-37). Each icon represents a corresponding object class that is available for use by developer.

20. It would have been obvious to one of ordinary skill in the art at the time of the invention was to add more graphical icons representing other programming objects which will improve the functionality of the system.

21. As per claim 14, it is similar to claim 13 with added symbols; therefore, it is rejected for the same rationale as claim 13.

22. As per claim 15, Bailey teaches the method, wherein the sixth, eighth, and ninth symbols are drawn connecting two other object symbols (col. 4, lines 35-40).

23. As per claim 16, it is similar to claim 13 with added symbols; therefore it is rejected for the same rationale as claim 13.

24. As per claim 17, Bailey teaches the method, further comprising the step of: providing in a separate description of the logic to be performed responsive to an event script (col. 10, lines 9-12).

25. As per claim 18, Bailey teaches the method, wherein the symbol representing event script type objects is drawn connected to another object that directly executes the event script corresponding to the event script symbol (col. 9, lines 60-62).

26. As per claim 19, Bailey teaches the method, wherein the symbol representing method type objects is drawn connected to the main object of the diagram (col. 4, lines 35-40) and represents that the object is available within that main object and does not represent that the main object invokes it (col. 7, lines 57-60; col. 8, lines 14-18).

Art Unit: 2193

27. As per claim 20, Bailey teaches the method, wherein the method is implemented via a computer program, and wherein step (1) comprises providing a graphical user interface in which a user is presented with a pallet containing the symbols (col. 7, lines 57-61).

28. However, Bailey does not explicitly teach wherein steps (3) and (4) comprise dragging and dropping the symbols from the pallet into a work area.

29. Kodosky in analogous art teaches computer executable instructions that enable the user to drag and drop symbols from the pallet into a workspace (paragraph [0037]; [0034] lines 6-9).

30. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Bailey and Kodosky so that user or developer can easily create, distribute and/or deploy application among various components in a distributed system.

31. As per claim 21, Bailey teaches the method, wherein the method is implemented via a computer program, and wherein step (1) comprises providing a graphical user interface in which a user is presented with a pallet containing the symbols (col. 7, lines 57-61) and wherein steps (3) and (4) comprise dragging and dropping the symbols from

the pallet into a work area (col. 12, lines 35-37), and wherein the labels comprise text (col. 10, lines 53-55) and further wherein at least some of the text labels can be made to appear in the graphical representation via an action taken by a user (col. 10, lines 63-65).

32. As per claim 22, Bailey teaches the invention substantially as claimed including a computer readable product embodied on computer readable media readable by a computing device for enabling a user to generate a graphical representation of object oriented programming logic (Abstract lines 10-13; col. 3, lines 24-27), the product comprising:

a graphical user interface in which a user is presented with a plurality of different symbols for use in developing a graphical representation of object oriented programming logic, each different symbol representing a different type of object in object oriented programming (col. 7, lines 57-61);

label the symbol with a label descriptive of the object's features (col. 9, lines 50-52); and

computer executable instructions that enable the user to draw lines between objects in the workspace (col. 4, lines 35-40; col. 9, lines 60-62).

33. However, Bailey does not explicitly teach computer executable instructions that enable the user to drag and drop symbols into a workspace.

Art Unit: 2193

34. Kodosky in analogous art teaches computer executable instructions that enable the user to drag and drop symbols into a workspace (paragraph [0037]; [0034] lines 6-9).

35. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Bailey and Kodosky so that user or developer can easily create, distribute and/or deploy application among various components in a distributed system.

36. As per claim 24, Kodosky teaches, fifth computer executable instructions that enable the user to denote one and only one object in the workspace as a main object (paragraph [0012] lines 9-11, 24-25).

37. As per claim 26, Bailey teaches the method, wherein the labels are text labels (col. 10, lines 53-55).

38. As per claim 27, Bailey teaches the method, further comprising: computer readable instructions that enable the user to prepare a plurality of the diagrams corresponding to separate parts of an overall application and further comprising computer readable instructions for enabling the user to specify relationships between individual ones of the diagrams (col. 8, lines 14-18; col. 9, lines 53-57).

39. As per claim 28, Kodosky teaches, wherein the sixth computer readable instructions comprise instructions that enable the user to including references associated with symbols in one diagram identifying at least one other diagram within which the object represented by that symbol also appears (paragraph [0015]).

40. As per claim 29, Kodosky teaches the method, wherein the sixth computer readable instructions comprise instructions that enable the user to specify in a first one of the diagrams the nature of the relationship of the representation of the object in the first diagram relative to the representation of the object in a second diagram (paragraph [0012] lines 1-5), wherein the relationship between the object as represented in the first and second diagrams is selected from the group comprising: (1) the second diagram discloses additional details about the object in the first diagram; (2) the second diagram shows the object in a more abstract context than the first diagram (paragraph [0012] lines 5-16); and (3) the object is the main object of the second diagram (paragraph [0012] lines 20-27).

41. As per claim 30, Bailey does not explicitly teach wherein the symbols representing different object types include: a symbol for representing objects that are application type objects; a symbol for representing objects that are window type objects; a symbol for representing objects that are class type objects; a symbol for representing

objects that are event script type objects; and a symbol for representing objects that are method type objects.

42. Bailey does teach plurality of objects or icons representing text boxes, radio buttons, scroll bars, menu bars and so on (col. 2, lines 3-9; col. 7, lines 57-61; col. 8, lines 8-11, 15-18, 24-37). Each icon represents a corresponding object class that is available for use by developer.

43. It would have been obvious to one of ordinary skill in the art at the time of the invention was to add more graphical icons representing other programming objects which will improve the functionality of the system.

44. As per claim 31, Bailey does not explicitly teach the method, wherein the symbols representing different object types include: a first symbol for representing objects that are application type objects; a second symbol for representing objects that are window type objects; a third symbol for representing objects that are class type objects; a fourth symbol for representing objects that are event script type objects; and a fifth symbol for representing objects that are method type objects; and wherein the additional symbols representing additional program elements include: a sixth symbol for representing data transfers; a seventh symbol for representing databases; an eighth symbol for representing remote links; and a ninth symbol for representing inheritance.

45. Bailey does teach plurality of objects or icons representing text boxes, radio buttons, scroll bars, menu bars and so on (col. 2, lines 3-9; col. 7, lines 57-61; col. 8,

lines 8-11, 15-18, 24-37). Each icon represents a corresponding object class that is available for use by developer.

46. It would have been obvious to one of ordinary skill in the art at the time of the invention was to add more graphical icons representing other programming objects which will improve the functionality of the system.

47. As per claim 32, Bailey teaches the method, seventh computer executable instructions that restrict the user to using the sixth, eighth, and ninth symbols to connect two other object symbols (col. 4, lines 35-40).

48. As per claim 33, it is similar to claim 30 with added symbols; therefore, it is rejected for the same rationale as claim 30.

49. As per claim 34, Bailey teaches the method, further comprising:

computer executable instructions that enable the user to providing in a separate description of the logic to be performed responsive to an event script (col. 9, lines 9-12).

50. As per claim 35, Bailey teaches the method, further comprising: computer executable instructions that enable the user to insert text associated with symbols in the

Art Unit: 2193

workspace that can be made to appear in the workspace responsive to an action taken by a user (col. 10, lines 9-12, 53-55, 58-59, 64-65; col. 34, lines 16-20).

51. Claims 3, 4, 23 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bailey (US Pat. No. 6,701,513) in view of Kodosky (US PG-Pub. No. 2003/0184580) further in view of Visio 2000 Standard Edition User Guide (Published by Visio International 1999 hereinafter Visio).

52. As per claim 3, Kodosky teaches the method, further comprising the step of: graphically denoting the main object in the diagram (paragraph [0012] lines 9-11, 24-25). But fails to teach drawing another symbol around the symbol for the main object.

53. Visio teaches drawing another symbol around the symbol for the main object (pages 15-17).

54. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Kodosky and Visio such that user or developer can easily create, distribute and/or deploy application with identifying various components in a distributed system as a main component in the system.

Art Unit: 2193

55. As per claim 4, Visio teaches the method, wherein step (7) comprises drawing a circle completely enclosing the symbol of the main object (page 18, see shape-to-shape connections).

56. As per claims 23 and 25, they are the product claims of claims 3 and 4, therefore; they are rejected for the same reason as per claims 3 and 4 above.

Response to Arguments

57. Applicant's arguments with respect to claims 1-35 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Loren Chauhan whose telephone number is 571-270-1554. The examiner can normally be reached on Mon.-Thr. 9:30-5:00 (EST).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on 571-272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2193

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Lewis A. Bullock, Jr./
Supervisory Patent Examiner, Art Unit 2193

/Loren Chauhan/
Examiner, Art Unit 2193